



Programa Institucional de Bolsas de Iniciação Científica – PIBIC
RELATÓRIO FINAL

IDENTIFICAÇÃO DO PROJETO

Título do Projeto: _____

Local de Realização (Unidade/Instituto/Departamento/Laboratório): _____

Endereço: _____

Bairro: _____ Cidade: _____ UF: _____ CEP: _____

DADOS DO ORIENTADOR

Nome: _____

Matrícula Siape: _____ CPF: _____

Endereço: _____

Bairro: _____ Cidade: _____

UF: _____ CEP: _____ E-mail: _____ - _____

Telefone 1: (____) _____ Telefone 2: (____) _____

DADOS DO BOLSISTA

Nome: _____

Matrícula: _____ CPF: _____ CR: _____

Curso/Departamento/Instituto: _____

Endereço: _____

Bairro: _____ Cidade: _____

UF: _____ CEP: _____ E-mail: _____

Telefone 1: (____) _____ Telefone 2: (____) _____

O RELATÓRIO científico FINAL DEVERÁ CONTER **TODOS** OS ITENS ABAIXO:

1. TÍTULO: Fonte Times New Roman 12 bold.

título: Usando problemas de minimização para otimizar circuitos eletrônicos usando Python com Algoritmos Evolutivos

palavras chave: Algoritmos Evolutivos, IA Generativa, Circuitos Eletrônicos

2. INTRODUÇÃO (descrever o tema abordado e os objetivos): Fonte Times New Roman 11 espaçamento 1,5.

3 METODOLOGIA (descrever de maneira clara e objetiva como foi realizada a pesquisa, as etapas desenvolvidas e descrever os materiais e métodos utilizados): Fonte Times New Roman 11 espaçamento 1,5.

Benefícios da Manutenção Preditiva com Machine Learning:

- **Redução de custos:** Evita paradas inesperadas na produção e custos de reparo emergenciais.
- **Aumento da produtividade:** Mantém os equipamentos em operação por mais tempo.
- **Melhoria da segurança:** Reduz o risco de acidentes e falhas catastróficas.
- **Otimização da programação de manutenção:** Permite o planejamento estratégico da manutenção, evitando sobrecarga de trabalho e otimizando recursos.
- **Tomada de decisões baseadas em dados:** Fornece informações valiosas para a tomada de decisões estratégicas sobre investimentos em equipamentos e otimização de processo

5) Programação em Python

Algoritmos de Machine Learning para Manutenção Preditiva: Previsão de Falhas com Precisão e Eficiência

A **Manutenção Preditiva** é uma ferramenta crucial para a indústria moderna, permitindo a antecipação de falhas em equipamentos antes que elas ocorram. Isso evita paradas inesperadas na produção, reduz custos de reparo e manutenção, aumenta a confiabilidade do sistema e garante a segurança dos trabalhadores.

Algoritmos de Machine Learning se destacam como aliados poderosos na Manutenção Preditiva, oferecendo soluções inteligentes para a análise de dados de sensores e histórico de falhas. Através de técnicas de aprendizado de máquina, podemos:

•

Prever falhas com antecedência:

•

- **Regressão Linear:** Um modelo clássico e robusto para prever valores contínuos, como a vida útil restante de um equipamento.
- **Regressão com Árvore de Decisão:** Uma abordagem flexível que captura relações complexas entre variáveis e permite a interpretação dos resultados.
- **Redes Neurais Artificiais:** Modelos poderosos que aprendem padrões não lineares nos dados, alcançando alta precisão em previsões complexas.

•

Classificar falhas com precisão:

- **K-Nearest Neighbors (KNN):** Um algoritmo simples e eficiente para classificar novas falhas com base em sua similaridade com falhas do passado.
- **Naive Bayes:** Um modelo probabilístico que classifica falhas com base na probabilidade de cada classe, considerando a independência entre as variáveis.
- **Máquinas de Suporte Vetorial (SVM):** Um algoritmo robusto para encontrar o hiperplano que melhor separa as classes de falhas, mesmo com dados complexos e não lineares.

Modelo Híbrido: Combinando o Melhor de Cada Mundo

Para otimizar a precisão e a eficiência da Manutenção Preditiva, podemos combinar modelos de regressão e classificação em um **modelo híbrido**. Essa abordagem oferece diversas vantagens:

- **Previsão precisa da vida útil restante:** O modelo de regressão prevê a vida útil restante do equipamento com alta precisão, permitindo o planejamento estratégico da manutenção.
- **Classificação precisa de falhas:** O modelo de classificação identifica o tipo de falha que está prestes a ocorrer, direcionando os esforços de manutenção para o problema específico.
- **Interpretação dos resultados:** A combinação de modelos permite uma melhor compreensão dos fatores que influenciam a falha dos equipamentos, auxiliando na tomada de decisões estratégicas.

algoritmos evolutivos deap

programação em flutter

Benefícios do sistema:

- **Otimização da produção:** A análise visual e a otimização em tempo real permitem reduzir custos e aumentar a eficiência da produção.
- **Controle de estoques preciso:** O monitoramento constante do estoque garante a disponibilidade de produtos e evita rupturas.
- **Tomada de decisões estratégica:** Dados atualizados e análises aprofundadas auxiliam na tomada de decisões estratégicas para a empresa.
- **Experiência do usuário aprimorada:** A interface intuitiva e a comunicação em tempo real garantem uma ótima experiência para os usuários.

rastrigin com scipy

metaheurísticas para problemas de minimização

1. Otimização do Layout da Fábrica:

- Algoritmos evolutivos podem otimizar o layout da fábrica para minimizar o tempo de deslocamento de materiais e produtos, aumentando a eficiência da produção.

- Metaheurísticas como Simulated Annealing ou Tabu Search podem encontrar soluções de layout robustas e eficientes, considerando diversos fatores como tamanho das máquinas, fluxo de materiais e restrições espaciais.

2. Agendamento de Produção:

- Algoritmos genéticos podem otimizar o agendamento de produção para minimizar tempos de espera, maximizar a utilização de recursos e reduzir custos.
- A capacidade de lidar com problemas complexos e multidimensionais torna os algoritmos genéticos ideais para otimizar o agendamento em cenários com diversas máquinas, produtos e restrições.

3. Otimização do Estoque:

- Metaheurísticas como Particle Swarm Optimization (PSO) podem otimizar os níveis de estoque para minimizar custos de armazenamento e rupturas de estoque.
- O PSO considera diversos fatores como demanda, lead time e custos de armazenamento para encontrar o equilíbrio ideal entre os níveis de estoque, evitando excessos e desabastecimentos.

4. Manutenção Preditiva:

- Algoritmos de aprendizado de máquina podem analisar dados de sensores e histórico de falhas para prever falhas em equipamentos e otimizar a programação de manutenção.
- Essa abordagem proativa evita paradas inesperadas na produção, reduzindo custos e aumentando a confiabilidade do sistema.

5. Logística e Distribuição:

- Metaheurísticas como o Algoritmo de Colônia de Formigas (ACO) podem otimizar rotas de entrega para minimizar tempo e custos de transporte.
- O ACO imita o comportamento das formigas em busca de alimento para encontrar rotas eficientes que consideram fatores como distância, congestionamento e tempo de carregamento/descarga.

Integração com o Sistema Existente:

- As metaheurísticas e algoritmos evolutivos podem ser integrados ao sistema existente através de APIs ou módulos específicos.
- A comunicação entre o sistema e os algoritmos de otimização pode ser realizada em tempo real ou de forma periódica, de acordo com as necessidades da empresa.

Benefícios da Aplicação:

- **Redução de custos:** Otimizando o layout, agendamento, estoque, manutenção e logística, a empresa pode reduzir custos de produção, armazenamento, transporte e manutenção.
- **Aumento da eficiência:** A otimização dos processos leva a um aumento da eficiência da produção, reduzindo tempos de espera, otimizando a utilização de recursos e diminuindo o retrabalho.
- **Melhoria da qualidade:** A otimização do processo de produção e a manutenção preditiva contribuem para a melhoria da qualidade dos produtos.
- **Maior agilidade e flexibilidade:** Os algoritmos de otimização permitem que a empresa se adapte rapidamente às mudanças de demanda, mercado e condições operacionais.
- **Tomada de decisões estratégica:** Dados otimizados e análises aprofundadas auxiliam na tomada de decisões estratégicas para a empresa.

4. RESULTADOS (apresentar os dados coletados e o tratamento dos mesmos. Nesta parte podem ser apresentadas as tabelas, cálculos, gráficos, fotos, etc): Fonte Times New Roman 11 espaçamento 1,5.

minimização global rastrigin graficos

5. PRODUÇÃO TÉCNICO-CIENTÍFICA (informar participações em seminários, cursos, participação em encontros científicos, palestras, etc, desenvolvidas durante o período considerado): Fonte Times New Roman 11 espaçamento 1,5.

Algoritmo evolutivos

o Algoritmo de Colônia de Formigas (ACO) que você mencionou é um tipo de **algoritmo evolutivo**. Ele se inspira no comportamento das formigas em busca de alimento para otimizar soluções em diversos problemas, como rotas de entrega, agendamento de tarefas e planejamento de produção.

Características dos Algoritmos Evolutivos:

- **Inspiração na natureza:** Imitam processos naturais, como seleção natural, mutação e recombinação, para buscar soluções de forma eficiente.
- **População de soluções:** Trabalham com um conjunto de soluções candidatas, explorando o espaço de busca de forma iterativa.
- **Operadores evolutivos:** Aplicam operadores como seleção, mutação e recombinação para modificar as soluções ao longo das gerações.
- **Busca por soluções otimizadas:** Visam encontrar a melhor solução possível para o problema em questão, considerando diversos critérios.

Diferenças entre ACO e Algoritmos Genéticos:

Embora ambos sejam algoritmos evolutivos, o ACO e os Algoritmos Genéticos (AGs) apresentam algumas diferenças:

- **Representação das soluções:** No ACO, as soluções são representadas por rotas ou sequências de elementos, enquanto nos AGs, as soluções são codificadas em cromossomos binários.
- **Operadores evolutivos:** No ACO, os operadores de seleção, deposição de feromônio e evaporação simulam o comportamento das formigas, enquanto nos AGs, os operadores de seleção, crossover e mutação modificam diretamente os cromossomos.
- **Mecanismo de busca:** No ACO, a busca por soluções é guiada pela deposição de feromônio, que indica caminhos mais promissores, enquanto nos AGs, a busca é guiada pela seleção de soluções mais aptas e pela aplicação de operadores de recombinação e mutação.

Vantagens do ACO:

- **Eficiência na busca:** O ACO pode encontrar soluções boas ou ótimas em tempo razoável, mesmo para problemas complexos com muitas variáveis.
- **Robustez:** O ACO é menos sensível a ruídos e incertezas nos dados, tornando-o adequado para problemas com informações incompletas ou imprecisas.

- **Facilidade de implementação:** O ACO pode ser implementado em linguagens de programação de alto nível, como Python e R, facilitando o desenvolvimento e a aplicação.

Aplicações do ACO:

- **Otimização de rotas de entrega:** Encontrar o caminho mais curto ou eficiente para entregar produtos ou serviços.
- **Agendamento de tarefas:** Definir a ordem ideal para executar tarefas em um processo, minimizando tempo de espera e otimizando recursos.
- **Planejamento de produção:** Otimizar a produção de bens ou serviços, considerando custos, tempo e qualidade.
- **Outras aplicações:** Resolução de problemas de TSP (Traveling Salesman Problem), otimização de redes de comunicação, balanceamento de carga em sistemas computacionais, entre outros.

metaheurísticas faculdade do ceará

tese doutorado rainer
RCE ALGORITMOS EVOLUTIVOS

`newCriterio` **Function:**

The `newCriterio` function implements a two-step approach for selecting elite individuals:

-

Step 1: Identify the Best Fitness and Maximum Difference:

-

a. Find the best individual (HOF) from the current population. b. Calculate the maximum difference allowed for fitness values, based on the `porcentagem` parameter.

-

-

Step 2: Select Elite Individuals Based on Fitness and Diversity:

-

a. Iterate through the population: i. Check if the individual's fitness is within the maximum difference range. ii. If so, calculate the difference between its decision variables and those of the previously selected elite individuals. iii. If the difference exceeds the `delta threshold`, consider it an elite individual. iv. Add the elite individual to the `pop_RCE` list and the `CONJUNTO_ELITE_RCE` set.

-

2. Applying RCE Periodically:

- The `run` method now includes a check for RCE application based on the `num_repopulation` parameter and the current generation.

- If RCE is enabled: a. Call the `aplicar_RCE` function to generate a new population using the `newCriterio` function for elite selection. b. Replace the current population with the modified population generated by RCE.

3. Additional Modifications:

- The `checkDecisionVariablesAndFitnessFunction` method now accepts `decision_variables` and `fitness_function` arguments.
- The `elitismoSimples` function updates the HOF and clones the best individual to the top of the population.
- The `criterio2` function compares decision variables and selects elite individuals based on the delta threshold.
- The `avaliarFitnessIndividuos` function evaluates the fitness of all individuals in the population.
- The `calculateFitnessGeneration` function calculates the fitness for the new population.
- The `visualizarPopAtual` function updates the `allIndividualValuesArray` with data from the current population.
- The `cout` function adds a decorative border to printed messages.

Key Improvements:

- The `newCriterio` function provides a more flexible and efficient approach for selecting elite individuals, considering both fitness and diversity.
- Applying RCE periodically allows for the introduction of new genetic material and exploration of diverse regions of the search space.
- The updated code includes additional helper functions and improved data handling.

Código Python para Otimização com Algoritmo Evolutivo e RCE

Este código implementa um algoritmo evolutivo com Estratégia de Reposição por Diversidade de Cadeia (RCE) para otimização de funções.

Bibliotecas utilizadas:

- `numpy`
- `math`
- `deap`
- `random`
- `matplotlib.pyplot`
- `time`
- `scipy.optimize`
- `json`
- `pandas`

Classe `AlgoritmoEvolutivoRCE`:

- Inicializa o algoritmo com parâmetros de configuração (`setup`).
- Define estatísticas de acompanhamento (`stats`).
- Armazena o logbook (`logbook`) para registrar o histórico de gerações.
- Cria a população inicial (`POPULATION`) e a população de otimização (opcional: `POP_OPTIMIZATION`).
- Armazena dados do RCE (`pop_RCE`, `best_solutions_array`, `best_individual_array`, `allIndividualValuesArray`, `data`, `repopulation_counter`, `allFitnessValues`, `CONJUNTO_ELITE_RCE`).
- Define métodos para:

- Registrar estatísticas e melhores soluções (`registrarDados`).
- Verificar clones na população (`checkClonesInPop`).
- Gerar informações de indivíduos (`generateInfoIndividual`).
- Exibir dataframe com informações dos indivíduos (`show_ind_df`).
- Aplicar os critérios RCE para selecionar indivíduos (`critério1, critério2_alternative, newCritério`).
- Realizar o RCE para gerar uma nova população (`aplicar_RCE`).
- Implementar elitismo simples (`elitismoSimples`).
- Avaliar o fitness dos indivíduos (`avaliarFitnessIndividuos`).
- Calcular o fitness da geração (`calculateFitnessGeneration`).
- Verificar variáveis de decisão e função fitness (`checkDecisionVariablesAndFitnessFunction`).
- Executar o algoritmo principal (`run`).
- Visualizar a população atual (`visualizarPopAtual`).
- Imprimir mensagens (`cout`).

Função `load_params`:

- Carrega parâmetros de configuração de um arquivo JSON.

Como utilizar o código:

1. Configure os parâmetros de otimização no objeto `setup`.
2. Defina as variáveis de decisão e a função fitness (opcionalmente, use `set_decision_variables_and_fitness_function`).
3. Execute o algoritmo (`algoritmo.run()`).
4. O algoritmo retornará a população final (`pop`), o logbook (`logbook`) e o melhor indivíduo (`hof[0]`).

Otimização com arduino e pandapower com python

A Física no Projeto Fluxo Ótimo de Potência (OPF) com Algoritmo Genético e Simulador de Rede Elétrica Simplificado

A física desempenha um papel fundamental no projeto **Fluxo Ótimo de Potência (OPF) com Algoritmo Genético e Simulador de Rede Elétrica Simplificada**. Os conceitos e equações da física elétrica são utilizados para modelar o comportamento da rede elétrica simplificada e otimizar o fluxo de potência.

Conceitos Físicos Aplicados:

- 1.

Leis de Kirchhoff:

- 2.

1. **Lei de Kirchhoff das Correntes (KCL):** A soma das correntes que entram em um nó é igual à soma das correntes que saem do nó.
2. **Lei de Kirchhoff das Tensões (KVL):** A soma das tensões ao longo de qualquer loop fechado em um circuito é igual a zero.
- 3.

Lei de Ohm:

4.

1. A tensão em um resistor é proporcional à corrente que o atravessa e à sua resistência.

5.

Perda de Potência em um Resistor:

6.

1. A potência dissipada em um resistor é proporcional ao quadrado da corrente que o atravessa e à sua resistência.

7.

Modelo de Transformador Ideal:

8.

1. A relação entre as tensões primárias e secundárias de um transformador ideal é igual à razão dos números de voltas nas bobinas.
2. A corrente primária de um transformador ideal é inversamente proporcional à corrente secundária.

Equações Relevantes:

1.

Equação da Corrente:

2.

1. $I = V / R$ (Lei de Ohm)

3.

Equação da Potência Dissipada:

4.

1. $P = I^2 * R$ (Perda de Potência em um Resistor)

5.

Equação da Relação de Tensão em Transformador Ideal:

6.

1. $V_p / V_s = N_p / N_s$

7.

Equação da Relação de Corrente em Transformador Ideal:

8.

$$1. \quad I_p = I_s * N_s / N_p$$

Modelagem Física no Pandapower:

O Pandapower utiliza equações físicas como as mencionadas acima para modelar o comportamento da rede elétrica simplificada. Os componentes da rede (geradores, transformadores, linhas de transmissão e cargas) são representados por modelos matemáticos que capturam suas características elétricas.

Otimização com Algoritmo Genético:

O algoritmo genético busca a configuração ideal da rede elétrica, minimizando a perda total de potência. Durante a otimização, o algoritmo considera as leis da física e as equações relevantes para avaliar cada configuração candidata.

Conclusão:

A física é essencial para compreender o funcionamento da rede elétrica simplificada e para otimizar o fluxo de potência. Os conceitos e equações da física elétrica fornecem a base para a simulação no Pandapower e para a otimização com o algoritmo genético.

Lembre-se: A física garante que o projeto OPF modele com precisão o comportamento da rede elétrica e encontre soluções otimizadas para o fluxo de potência.

6. CONCLUSÕES (Tomando como base a discussão dos resultados, concluir se os objetivos propostos foram alcançados): Fonte Times New Roman 11 espaçamento 1,5.

rastrigin global
sistema de potencia pandapower

7. REFERÊNCIAS BIBLIOGRÁFICAS (apresentar as fontes bibliográficas consultadas. Consultar as normas da ABNT): Fonte Times New Roman 11 espaçamento 1,5.

8. AUTO-AVALIAÇÃO DO ALUNO (o aluno deve avaliar sua participação no programa): Fonte Times New Roman 11 espaçamento 1,5.

O relatório final poderá conter até 40 páginas.

